

RTMaps for Wifibot Software Development Kit Quickstart

Content

Introduction.....	1
Folder tree	1
The SDK Wizards for Visual C++ 2005 and 2008 (My first RTMaps 4 Wifibot package)	2

Introduction

This Software Development Kit will allow a C++ developer to write and compile packages of components to run on the Wifibot robot under the RTMaps 4 Wifibot environment.

Note: This SDK can be installed on any computer, however note that package files (.pck) compiled under this SDK will not work if loaded on a standard RTMaps version. On the other hand, if you already have a standard RTMaps version, a package compiled with the standard RTMaps SDK will work on RTMaps 4 Wifibot.

Reminder: Limitations of RTMaps 4 Wifibot compared to the standard RTMaps release relies in that RTMaps 4 Wifibot does not provide the record and playback functionalities (Recorder and Player components, plus related record/replay methods).

Folder tree

Here is a quick overview of the RTMaps 4 Wifibot SDK folder tree:

[SDK_PATH] /

In the root folder of the RTMaps 4 Wifibot SDK installation, you can find the Visual C++ solution files (.sln) for the RTMaps 4 Wifibot SDK. Start here...

- bin/
Contains the SDK wizards for Visual C++ 2005 and 2008. These are the 2 Visual Basic Script files (.vbs). Double-click the one you need to automatically create a project for a RTMaps 4 Wifibot package or a component inside an existing package.
- doc/
Contains the documentation files (including the document you are currently reading... so I guess you guessed).
- include/
Contains the headers files for the RTMaps 4 Wifibot API (the main one is maps .hpp)

- `lib/`
Contains the lib files for the ^{RT}Maps 4 Wifibot API (the main one is `rtmaps4wifibot_pck.lib`)
- `packages/`
This is where your own compiled packages (.pck files) will be generated.
- `share/`
Don't touch. These are template files used by the SDK wizards and generated projects.
- `src/`
This is where you own source codes will be. You should not access these folders directly since project files and code skeletons can be generated automatically with the Visual C++ Wizards (.vbs files in the `bin/` folder).
 - `samples.u/`
A project of package with a few sample component source codes for demonstrating different generic features of the ^{RT}Maps 4 Wifibot SDK.
 - `wifibot.u/`
You'll find here the source codes and project of the components dedicated to the wifibot robot. These are used for control and communication with the wifibot. The source codes are provide so that if you want to adapt them... feel free!
 - `[my_package.u]/`
One of your own project and source files for your own package (`my_package.pck`).

The SDK Wizards for Visual C++ 2005 and 2008 (My first ^{RT}Maps 4 Wifibot package)

First of all, open the solution file in the SDK root folder corresponding to your Visual C++ version (2005 or 2008).

You can see 2 projects already available with a few components inside:

- `rtmaps_samples`: a number of components providing examples of source codes demonstrating the generic capabilities of ^{RT}Maps 4 Wifibot components.
- `rtmaps_wifibot`: a few components for controlling and communicating with the wifibot robot that are already provided in binary form in your ^{RT}Maps 4 Wifibot installation, but in case you want to adapt them, you can find their sources here.

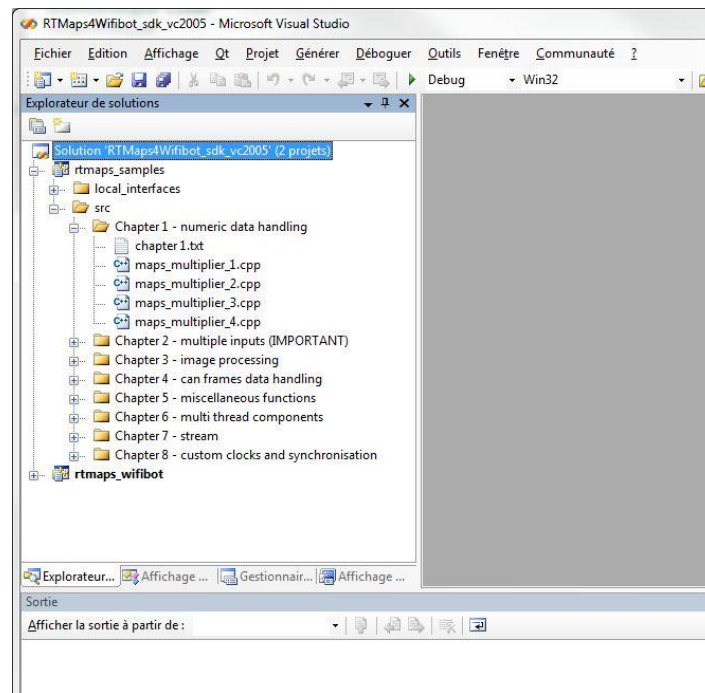


Figure 1 - The two projects available in the RTMaps 4 Wifibot solution after installation and in Visual C++ 2005

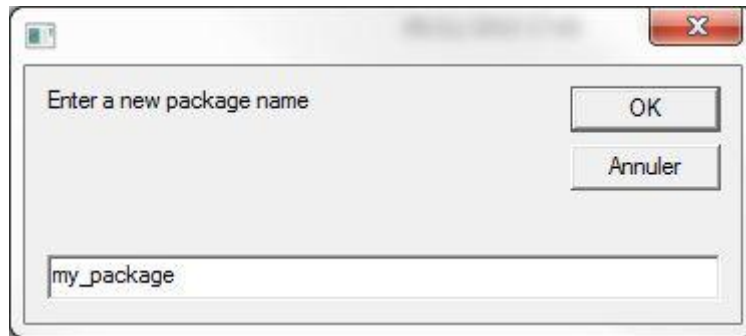
The wizards will now allow you to create your first package project and a first component to compile inside that project.

The wizards are the two .vbs files in the `bin/` folder.

Double-click on the one corresponding to your Visual C++ version. A window appears with 3 choices. Type '1' to create a new package, then "OK" (of course...)



Then a new box requires a package name. Enter the one you want, this is your project name, and it will end-up in the name of the future .pck file. Let's go with 'my_package'.



A folder called 'my_package.u' has been created in the `src/` folder, with a `.vcproj` file inside and a few subfolders. Let's not care about it for now.

Re-launch the wizards (the `.vbs` file).

In the first window, now type '2' to create a new component.

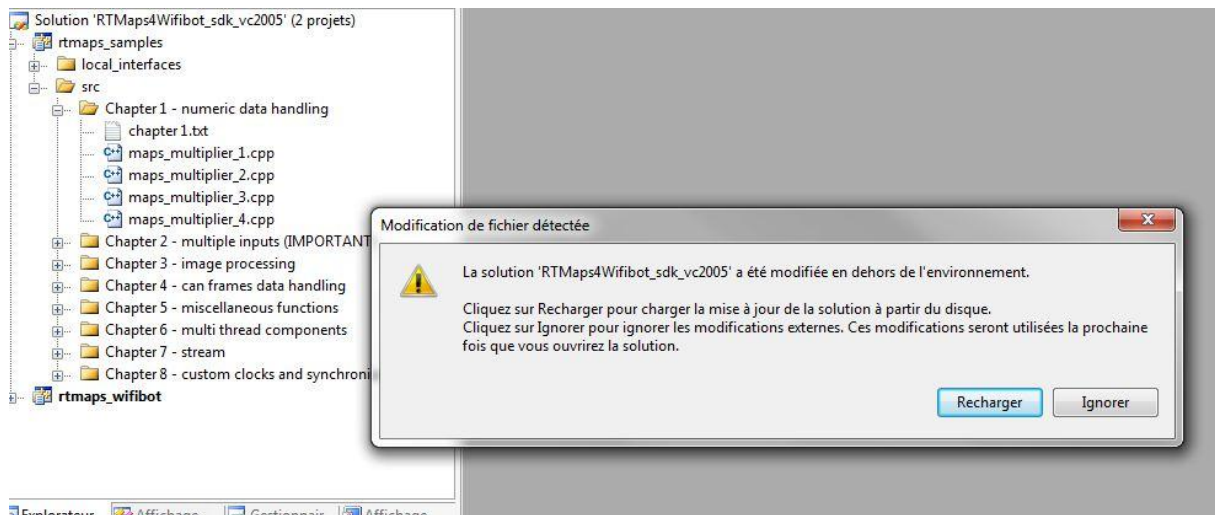
The next box requests the existing project you want your component to be created in. Type the index corresponding to the project 'my_package' (or the one you just created here above, whatever it is).

Then enter the name of your choice for your future component. This is how the component will be named once the package is loaded in the ^{RT}Maps 4 Wifibot environment on the robot. Let's go with 'my_component'.

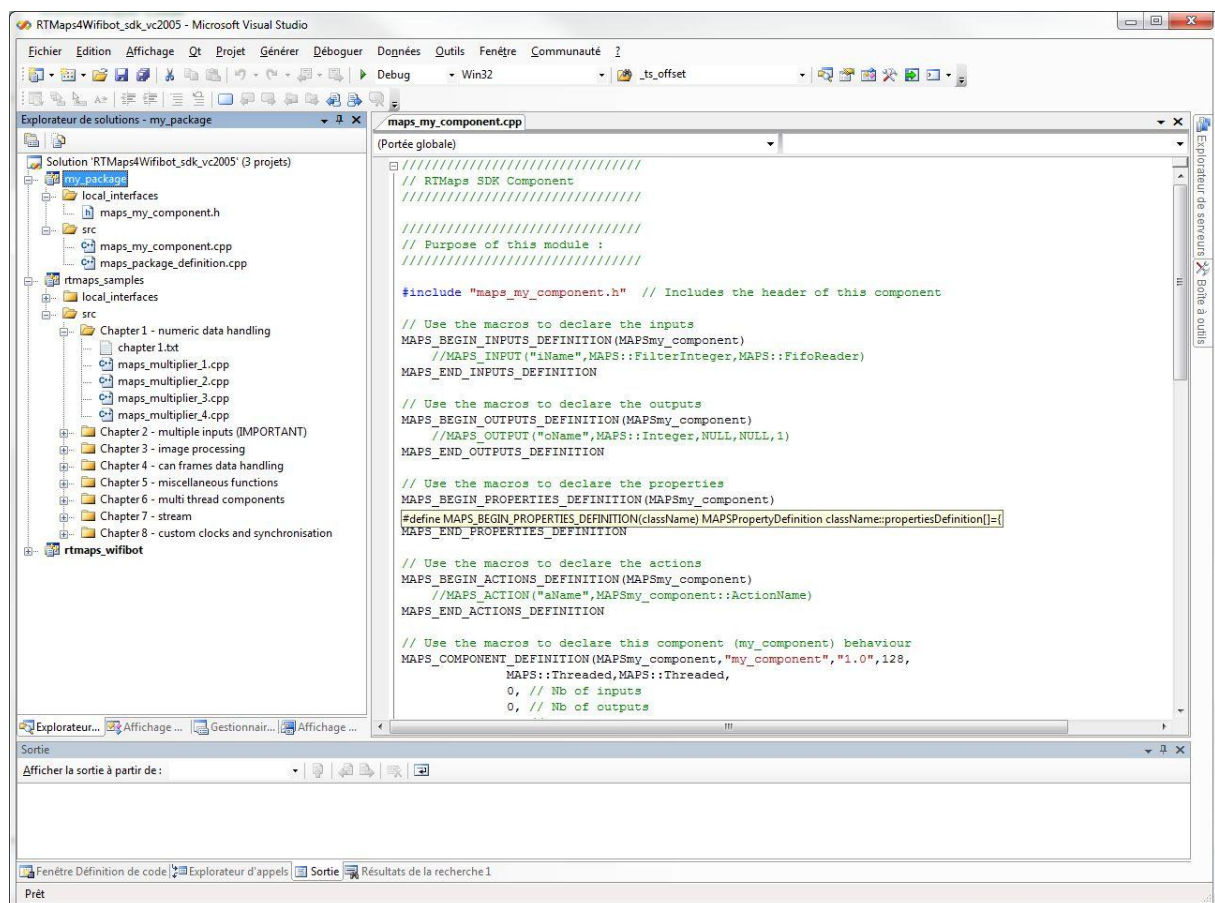


Two files (a `.cpp` and a `.h` file) have been generated in the `src/` and `local_interfaces/` subfolders of `my_package.u/`, itself in the `src/` folder of your ^{RT}Maps 4 Wifibot SDK installation.

Go back to your Visual C++ window. It says the current solution has been modified outside the environment. Click 'Reload'



Now you can see that the new 'my_package' project has been integrated in your solution. If you browse its content, you can check the availability of a first ^{RT}Maps 4 Wifibot component source files. You can compile the project as such and check it has been generated in the package/[debug/] folder of the ^{RT}Maps 4 Wifibot installation. You can now export this .pck onto your robot and load it in your ^{RT}Maps 4 Wifibot environment, then place the component on the diagram and see it working (it does not much at this point but prompting messages in the console).



You can now edit your component code and integrate your own methods and algorithms. This is another topic, read the `rtmaps_developer_manual.pdf` to learn more.

Enjoy your robot !